



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/628,738	07/28/2003	Naga A. Ayachitula	SVL920030044US1	4034
7590	11/21/2006		EXAMINER	
Michael E. Hudzinski FAY, SHARPE, FAGAN, MINNICH & MCKEE, LLP 1100 Superior Avenue, Seventh Floor Cleveland, OH 44110-2518			ROSE, HELENE ROBERTA	
			ART UNIT	PAPER NUMBER
			2163	
			DATE MAILED: 11/21/2006	

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	10/628,738	AYACHITULA ET AL.
	Examiner	Art Unit
	Hélène Rose	2163

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 13 September 2006.
- 2a) This action is FINAL. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-9 and 15-32 is/are pending in the application.
 - 4a) Of the above claim(s) 10-14 is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-9 and 15-32 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on 28 July 2003 is/are: a) accepted or b) objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 - a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|---|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

Detailed Action

1. In response to communication filed on 9/13/2006, Claims 1-32 are presently pending.
Claims 1, 9, 15, 20, 24, and 32 have been amended. No claims were added, nor cancelled.

2. Applicants arguments with respect to claims 1-32 have been considered, but are not persuasive.

Claim Rejections 35 U.S.C – 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-32 are rejected under 35 U.S.C. 103(a) as being unpatentable over anticipated by Waldo et al (US Patent No. 6,016,500, Date of Patent: January 18, 2000) in view of Kolodner et al (US Patent No. 6,675,379, Filing Date of Patent: June 30, 2000).

Claims 1 and 24:

Claims 1 and 24 disclose a method/article of manufacture utilizing the same functionality, wherein the claim language is the same.

Regarding claims 1 and 24, teaches a method/article of manufacture comprising a program storage medium readable by a computer and embodying one or more instructions by the computer to perform a method of releasing resources of a user session operating in a

Art Unit: 2163

software environment that includes automatic memory management algorithm, the method comprising:

detecting an impending execution of the automatic memory management algorithm (column 13, lines 36-45, wherein upon detection of a failure, both the client and the server invoke each other's failure recovery routine to perform failure recovery, if the server has experienced a failure, once the client detects the failure, the client invokes the server's recovery routine which will perform recovery on the server, for example, the recovery routine may restart the server and send a message to the system administrator and if the client experiences a failure, the server invokes the client's recovery routine, thus performing failure recovery on the client, Waldo);

responsive to the detecting (column 4, lines 52-57, Waldo), accessing an object of the user session (Figure 11, diagram 11002 and column 14, lines 16-19, wherein groups of storage locations may have many programs varying for access; lines 20-23, wherein a program request a lease from the file system manager, i.e. the server, to access the group of storage locations for a period of time; lines 28-29, wherein the client may access the group of storage locations for the duration of the lease period and lines 49-58, wherein access parameters determines the type of access the server supports for that storage location and column 17, lines 46-49, wherein the server accesses the Java space containing all objects if any received from clients as part of lease request, Waldo);

Waldo discloses all the limitations above. However, Waldo does not disclose **traversing an object graph**; nor does he disclose the steps of identifying one or more external resource references of said object; releasing said one or more external resource references; nor does he disclose repeating the accessing, identifying, and releasing for each object of the user session. On the other hand, Kolodner discloses **traversing an object graph** (Figure 6, all features,

wherein object graph is interpreted and defined to be a view of an object system at a particular point and column 10, lines 34-39, wherein java garbage collector at this stage of program would view the objects referenced by elements of array as reachable, Kolodner); identifying one or more external resource references of said object (column 5, line 22, wherein identifying in the program an array of array elements and column 4, lines 7-9, wherein the array elements comprise object references contained in an array of such references, Kolodner); releasing said one or more external resource references **by a set of rules for said object** (column 4, lines 63-64, wherein the tool is used in software testing, to verify that software code is free of memory leaks; and column 16, lines 35-42, wherein two rules are defined, Kolodner); and repeating the accessing, identifying, and releasing for each object of the user session (Figure 1, diagram 21 and column 9, lines 33-43, wherein a user session is defined, Kolodner). It would have been obvious to one of the ordinary skill in the art at the time of the invention to incorporate Kolodner teaching into Waldo system. Waldo and Kolodener are analogous art because they both relate to detecting and removal of memory leaks. A skilled artisan would have been motivated to combine as suggested by Kolodner a feature that was able to identify, release, access external resources for accuracy of data and space, efficient processing of data, as well as to improving the performance of system.

Claims 2 and 25:

Regarding claims 2 and 25, in combination of Waldo in view of Kolodner teaches performing the accessing, identifying, releasing, and repeating as a Listener method belonging to a Java MyListener class in a Java environment (Figure 4, diagram 21 and column 12, lines 17-19, Kolodner); and registering the Listener method with the user session (column 16, lines 47-53, Waldo).

Claims 3 and 26:

Regarding claims 3 and 26, in combination of Waldo in view of Kolodner teaches wherein the registering includes setting a session attribute to correspond to an instance of the Listener method (page 16, lines 53-59, Waldo).

Claims 4 and 27:

Regarding claims 4 and 27, in combination of Waldo in view of Kolodner teaches wherein the detecting includes notifying the registered Listener method of the impending expiration of the user session (page 17, lines 13-14, Waldo)

Claims 5 and 28:

Regarding claims 5 and 28, in combination of Waldo in view of Kolodner teaches wherein the detecting includes detecting an impending expiration of the user session (Figure 10, diagrams 10008, wherein if the client has not completed its use of the file, the client determines if the lease is about to expire, Waldo).

Claims 6 and 29:

Regarding claims 6, in combination of Waldo in view of Kolodner teaches wherein the accessing, identifying, releasing, and repeating is performed prior to the execution of the automatic memory management algorithm (column 4, lines 40-43, wherein at the end of the lease period the guarantee that the reference to the resource will continue lapses, Waldo).

Claims 7 and 30:

Regarding claims 7 and 30, in combination of Waldo in view of Kolodner teaches wherein the identifying includes identifying a file resource (Figure 10, diagram 10005 and column 17, lines 6-7, Waldo); and the releasing includes closing said file resource (Figure 10, diagram 10006, wherein done with file is defined, in which the yes, is interpreted to be closing

the file, Waldo)

Claims 8 and 31:

Regarding claims 8 and 31, in combination of Waldo in view of Kolodner teaches wherein the identifying includes identifying an allocated resource; and the releasing includes deallocating the allocated resource (Figure 10, all features, Waldo).

Claims 9 and 32:

Regarding claims 9 and 32, in combination of Waldo in view of Kolodner teaches wherein the accessing of an object of the user session includes obtaining an object identifier corresponding to said object from the object graph (column 5, lines 63-67, Kolodner); and retrieving said object using the object identifier (column 6, lines 54-59, Kolodner).

Claim 15:

Regarding claim 15, in combination of Waldo in view of Kolodner Waldo teaches a system comprising a software program configured to initiate (Figure 400, diagram 420, Waldo), process (column 4, lines 55-56, performs various processing with respect to a resource managed by server, Waldo), and terminate user sessions (column 1, lines 39-41, wherein failure to reclaim memory at the proper point may lead to memory leaks with unreclaimed memory accumulating until the program terminates or memory space is exhausted; column 14, lines 59-62, when the backward iteration has terminated; and column 17, lines 42-44, wherein termination of the algorithm is accelerated by widening, i.e. deliberately losing some information and column 4, lines 43-47, wherein the application holding the reference to the resource and the garbage collection system managing the resources and the garbage collection system managing the resource agree to a finite guaranteed lease period both can know when the lease and therefore the guaranteed, expires and lines 56-57, wherein the lease is about to expire, Waldo);

an object graph defining an interrelationship between objects of said user session

(column 5, lines 35-40, wherein finding the relation includes finding an inequality relationship governing possible values of the program variable at the given point in the program and wherein finding the inequality relationship, wherein inequality is equivalent to interrelationships, includes defining a constraint graph that determines a bound on permitted values of the program variable);

a resource deallocation module linked to the software program to deallocate allocated external resources of each object of a user session responsive to an impending termination of said user session (column 2, lines 32-40, wherein a resource management involves allocating resource, i.e. memory in response to request as well as deallocating resources at appropriate times and column 17, lines 51-55, wherein if there are objects in the java space it indicates that the server terminates processing due to a failure and must perform recovery, the server performs recovery by invoking the recover method for each client that had a lease, Waldo); and

an automatic memory management module invoked subsequent to the deallocation performed by the resource deallocation module (column 3, lines 14-25, wherein distributed garbage collection must maintain integrity between allocated resources and the references to those resources, in other words, the system must not be permitted to deallocate or free a resource when an application running on any computer in the network continues to refer to that resource, Waldo).

Claim 16:

Regarding claim 16, in combination of Waldo in view of Kolodner teaches a Java virtual machine implementing the software program (column 6, lines 5-7, wherein the MI component may consist of a number of software modules preferably written in JavaTM programming language, Waldo), the resource deallocation module (column 4, lines 38-46, wherein application

Art Unit: 2163

holding a reference to a resource and the garbage collection system managing that resource, agree that the resource and a reference to that resource will be guaranteed, at the end of the lease period, the guarantee that the reference to the resource will continue lapses, allowing the garbage collection system to reclaim the resource, because the application holding the reference to the resource and the garbage collection system managing the resource agree to a finite guaranteed lease period, which is equivalent to resource deallocation, wherein deallocation is defined to be releasing control of memory that was previously allocated), and the automatic memory management module (column 4, lines 38-40, wherein garbage collection is defined to be an automatic process which attempts to free memory that is storing garbage, Waldo).

Claim 17:

Regarding claim 17, in combination of Waldo in view of Kolodner teaches wherein the resource deallocation module includes a deallocation listener method adapted to deallocate the allocated external resources of each object of said user session responsive to a notification of the impending termination of said user session (column 3, lines 61-67, wherein such failures can take the form of computer or application failure or network failure that prevent delivery of messages notifying the garbage collection system that a resource is no longer being referenced and column 18, lines 11-31, wherein the server determines whether it has received a renew request from the client, if the renew request has been received, the server renews the lease, and if, however, a renew has not been received, the server determines if a cancel request has been received by the client invoking the cancel method, and if the client invoked the cancel method, the server cancels the lease by deleting the object stored in step 11010 from the Java space, and if this is the last outstanding lease on the file, the server deletes the file, and if a cancel request was not received, the server determines if the lease has expired and if the lease

Art Unit: 2163

has not expired, processing continues, wherein however, if the lease has expired, the server knows that a failure has occurred and therefore invokes the recover method on the object in the Java space for the client with the lease that terminated, Waldo).

Claim 18:

Regarding claim 18, in combination of Waldo in view of Kolodner teaches wherein the resource deallocation module is linked to the software program by registration of the deallocation listener method with said user session (Figure 1, all features, Kolodner).

Claim 19:

Regarding claim 19, in combination of Waldo in view of Kolodner teaches wherein the resource deallocation module is linked to the software program by an assignment of an attribute of said user session to the deallocation listener method (Figure 11, all features, Waldo).

Claim 20:

Regarding claim 20, in combination of Waldo in view of Kolodner wherein the resource deallocation module being adapted to access the object graph to identify the objects of the user session (columns 5-6, lines performing the analysis of program variable relations includes defining a constraint graph that determines a bound on permitted values of the program variable at the given point in the program, wherein permitted is equivalent to accessing, and performing the liveness analysis includes adding a constraint to the constraint graph with respect to liveness of the elements to which the program variables correspond, and finding the range of the elements includes performing a flow analysis of the program so as to identify one or more live ranges of the array, wherein an element within the one or more ranges is alive at the given point in the program if there is an execution sequence using the element in the program following the given point, and there is no assignment of the memory location assigned to the element intermediate the given point and the use of the element by the execution sequence,

wherein, identifying the one or more live ranges includes performing a data flow analysis of the program in a backward direction relative to the execution sequence, Kolodner).

Claim 21:

Regarding claim 21, in combination of Waldo in view of Kolodner teaches wherein the automatic memory management module is invoked by the software program to process a plurality of user sessions including said user session (column 16, lines 2-15, wherein data processing system is defined along with its parts and wherein the client request access to one or more of the files by requesting a lease from the server, Waldo).

Claim 22:

Regarding claim 22, in combination of Waldo in view of Kolodner teaches wherein the automatic memory management module is invoked by an operating system to process software including said software program that operate under said operating system (Figure 8, all feature, wherein it illustrates a client platform and a sever platform and column 3, lines 1-7, wherein distributed garbage collection describes a facility provided by a language or runtime system for distributed systems that automatically manages resources used by an application or group of application running on different computers in a network, Waldo).

Claim 23:

Regarding claim 23, in combination of Waldo in view of Kolodner teaches wherein the resource deallocation module is integrated with the automatic memory management module as a single unitary memory management unit that executes prior to the termination of said user session (page 17, lines 35-49, Waldo).

Response to Applicant

Examiner clarifies the oversight within the office action mailed on 6/5/2006, wherein Applicant's election with traverse of Group 1 and not without traverse.

Examiner Response

Applicant argues prior art fails to teach, "**accessing an object of the user session and traversing an object graph, identifying one or more external resource references of said object, releasing said one or more external resources references by a set of rules for said object, and repeating the accessing, identifying, and releasing for each object of the user session**"

Examiner respectfully disagrees. In response to, "accessing an object of the user session". Referring to Waldo, column 16, lines 10-13, wherein the client request access to one or more files by a requesting lease a lease from the server, wherein this is equivalent to accessing an object of the user session – wherein user session is defined to be a session of activity of a user, and wherein client is interpreted to be a computer hardware or software used by an end user on a computer network; column 16, lines 52-57, wherein the type of access requested indicates the type of storage location access the client requested, wherein the types of access includes read access, write access, allocation access, re-allocation access, and sub-lock access, wherein the privilege field indicates the privilege level of the user or the client.

In response to "traversing an object graph"; applicant argues new claim language, which was not presently defined within the original office action, therefore in response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., traversing an object graph) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations

from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Examiner respectfully disagrees. In response to, "identifying one or more external resource references of said object, releasing said one or more external resources references by a set of rules for said object". Referring to, Kolodner, column 5, line 22, wherein identifying in the program an array of array elements; column 4, lines 7-9, wherein the array elements comprise object references contained in an array of such references; and column 6, lines 21-24, wherein identifying elements in the array that are outside the range so that memory locations assigned thereto can be reclaimed by a garbage collection function, and wherein external resources are defined to be resources that are outside of the computational environment; and

In response to, "releasing said one or more external resources references by a set of rules for said object". Referring to Kolodner, column 4, lines 63-64, wherein the tool is used in software testing, to verify that software code is free of memory leaks; and column 16, lines 35-42, wherein two rules are defined.

Examiner respectfully disagrees. In response to, "repeating the accessing, identifying, and releasing for each object of the user session". Referring to Waldo, Figure 11, all features, wherein the figured is further defined within columns 17-18, lines 35-67 and lines 1-21, in which access the java space, store objects, if failure is detected, it may access the java space and perform recovery by invoking recovery methods on the objects, and after deleting the objects the server receives a least request from one of the clients, after receiving the least request the objects received in this request into java space, and by storing the objects persistently, if the

failure occurs, the sever can access the java space and invoke the recover method on the object, and wherein this is interpreted to be repeating the access.

Applicant argues prior art fails to teach, "***an object graph defining an interrelationship between objects of a user session, a resource deallocation module linked to the software program to deallocate allocated external resources of each object of a user session responsive to an impending termination of said user session, and an automatic memory management module invoked subsequent to the deallocation performed by the resource deallocation module***"

Examiner respectfully disagrees. In response to, "*an object graph defining an interrelationship between objects of a user session*". Referring to Kolodner, column 5, lines 35-40, wherein finding the relation includes finding an inequality relationship governing possible values of the program variable at the given point in the program and wherein finding the inequality relationship, wherein inequality is *equivalent* to interrelationships, includes defining a constraint graph that determines a bound on permitted values of the program variable.

Examiner respectfully disagrees. In response to "*a resource deallocation module linked to the software program to deallocate allocated external resources of each object of a user session responsive to an impending termination of said user session*". Referring to Waldo, column 2, lines 32-40, wherein a resource management involves allocating resource, i.e. memory, in response to request as well as deallocating resources at appropriate times and column 17, lines 51-55, wherein if there are objects in the java space it indicates that the server

terminates processing due to a failure and must perform recovery, the server performs recovery by invoking the recover method for each client that had a lease.

Examiner respectfully disagrees. In response to, "an automatic memory management module invoked subsequent to the deallocation performed by the resource deallocation module". Referring to Waldo, column 3, lines 14-25, wherein distributed garbage collection, wherein garbage collection is defined to be a form of automatic memory management, and must maintain integrity between allocated resources and the references to those resources; column 8, lines 20-35, wherein initiates a garbage collection cycle to reclaim resources for which it determines either no more references are being made to the resources and so forth, and wherein reclaim is interpreted to resources of memory that was previously allocated, i.e. granted.

Prior Art of Record

1. Kolodner et al (US Patent No. 6,675,379)
2. Marion et al (US Patent No. 6,782,462)

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO

MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Point of Contact

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Helene Rose whose telephone number is (571) 272-0749. The examiner can normally be reached on 8:00am - 4:30pm Monday-Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Don Wong can be reached on (571) 272-1834. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Helene Rose
Technology Center 2100
November 15, 2006



DON WONG
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100